Modeling and Evaluation of a Data Center Sovereignty with Software Failures

Shakthivelu Janardhanan
Chair of Communication Networks
Technical University of Munich
Munich, Germany
shakthivelu.janardhanan@tum.de

Carmen Mas-Machuca
Chair of Communication Networks
Technical University of Munich
Munich, Germany
cmas@tum.de

***Abstract*—Technology Sovereignty is the capacity of a state to provide a technology to its constituents by developing or outsourcing the technology without causing a dangerous dependency on a particular contributor. There are several laws to monitor and govern networks that ensure the privacy and security of the consumers. However, they do not control the infrastructure or hardware dependency associated with these networks. A robust end-to-end network requires reliable hardware, software, and network architecture to handle multiple failures, increased users, and increased traffic demands. To handle multiple failures efficiently, there must be little to no dependency on the manufacturers involved. Therefore, the robustness of a network requires an adequate degree of sovereignty. In this study, Data Center Networks (DCN) are considered to understand the impact of choosing the hardware and software manufacturers appropriately on the DCN robustness. The unavailability of a component from a manufacturer or the presence of a software bug in the switches of a DCN can affect several flows, leading to multiple failures and huge losses of data. Thus, it is rational to find the best combination of manufacturers to create a network with the least dependency on the manufacturer(s). In this research, the sovereignty of a DCN is evaluated by considering different multiple-failure scenarios based on hardware and software manufacturers' reliability. Design guidelines for DCN operators are provided based on the findings of the analyses.**

***Index Terms*—sovereignty, reliability, data center**

I. Introduction

Technology drives the economy of a state. The recent trade conflicts, sanctions, and economic crises have clearly shown that the economy of a state is largely dependent upon the decisions in other states. The absence of a limitation on this dependency paves the way for establishing 'Strategic Autonomy'. Strategic Autonomy is the ability of a state to adopt and carry forward its own policies without being swayed by other states. It is essential for a state that its core technologies stand untroubled by geopolitical issues.

However, constructing an indigenous technology from scratch without any foreign elements is cumbersome. Therefore, it is more pragmatic to establish a technology solution that raises little-to-no adverse structural dependencies on foreign elements. The digital economy of a state depends on the amount of critical information possessed, the ease of access, and security. Data centers are responsible for all the aforementioned aspects, and hence they make the cornerstones of the Information Age.

In recent times, there have been notable cases where geopolitical motivations have encouraged governments to stop using products from companies belonging to specific countries. For example, Huawei is a leading Chinese corporation that sells telecommunication equipment. However, in late 2020, the trade war between China and USA forced the U.S. Federal Communications Commission (FCC) [1] to ban Huawei equipment entirely from their networks. Similarly, the United Kingdom has ruled that the new 5G kit from Huawei must be removed from their telecom infrastructure [2]. Since the war in Ukraine, EU has also placed sanctions on goods that cannot be exported to Russia. Cutting-edge technology including quantum computers, advanced semiconductors, high-end electronics, and software are banned from being exported to Russia [3]. If a network consists of several components from a vendor that the government disapproves of, the network operator has a serious problem to deal with. In such cases, all the components from said vendor have to be replaced at the earliest.

Apart from geopolitical issues, the availability and reliability of the different sub-components of a product also play a crucial role. A classic example is the Samsung Galaxy Note 7 issue [4]. Note 7 units carried a battery with a design flaw that caused them to overheat and explode. Samsung had to officially recall 2.5 million Note 7 units in September 2016, and redesign the model with outsourced batteries. In this case, all the devices that used this specific sub-component had to be removed from the network at the same time. Instead of user equipment, if it had been switches exploding in the data center, the problem would be much bigger, leading to network disruption for several hours before the network could be restored.

Software issues also play a vital role in causing multiple failures and long down times. For example, Canada shutdown 4000 government websites in December 2021, due to a security vulnerability in the popular Java-based logging utility called Apache Log4j [5]. Another example is the infamous Equifax data breach that occurred between May and July 2017 [6]. Equifax is a consumer credit reporting company.

This work has been funded by the Bavarian Ministry of Economic Affairs, Regional Development, and Energy under the project "6G Future Lab Bavaria".

It used the open-source web application framework called Apache Struts. Equifax failed to update its servers with a critical security patch from Apache Struts immediately. This left its servers vulnerable to attacks from hackers. In this case, every server that used this third-party framework was a threat to their network. The records of over 150 million people were compromised as a result. Moreover, Y2K and Y2K38 are popular errors related to formatting of calendar data in computers [7].

From these examples, it is safe to say that a small issue from a software library or a hardware component can cause the shutdown of several components in a network and lead to significant down times. It is also important to note that these issues may affect multiple components at the same time. Unfavorable political policies may also warrant the shutdown and replacement of several components in the network. Even if components are not affected simultaneously, the components would have to be replaced simultaneously or within a very short interval of time. Hence, manufacturer reliability is vital to build a robust network. Both these cases invariably result in substantial economic losses.

In data centers, switches are purchased from different manufacturers. Generally, increasing redundancy by adding backup paths is a popular technique used to improve performance under stress. The presence of backup paths ensures connectivity in the network even if a few nodes fail. At first sight, this technique seems to decrease the dependency on manufacturer reliability. But, authors in [8] and other previous researches expose that this redundancy is not enough. The arrangement and interconnections of components based on the manufacturers' reliability also play an essential role in the performance of the DCN.

This paper investigates the worst-case scenarios, i.e., the impact of multiple failures based on the hardware (HW) and software (SW) manufacturers' reliability in a DCN. The unavailability of components, unavailability of parts of a component, misbehavior of hardware, and software bugs of varying severity are the prominent potential threats which may degrade the performance of a DCN. Therefore, in this paper, the focus lies on inspecting the effect of,

(P1) choosing the required number of manufacturers (N_m) for both HW and SW,
(P2) setting the upper bounds on the number of components from a single hardware manufacturer, and the number of components that can use software from a single manufacturer.

on the robustness and reliability of a DCN. This effect is evaluated numerically in terms of connectivity, and max-flow. This paper is a continuation of our previous work [9], which explored only HW failures.

The paper is structured as follows: Section II briefs the related work. Section III explains the DCN modeling. Section IV discusses the simulation details and the two methods adopted for the sovereignty analysis. Section V discloses the results. Section VI concludes this paper.

II. Related Work

The authors in [10] give a clear understanding of the term digital sovereignty. Although the ability to outsource a technology provides a better experience for the consumers of a state, it has some conflicting interests. When the provider of that technology is unavailable, the entire service is affected. The goal of sovereignty is to avoid such a dependency on a single contributor.

In [8], [11], [12], the authors use real-time information from data centers to study the root causes of failures. Hardware faults, misconfigurations, and software bugs are among the common root causes. In [8], [11], [13], [14], the authors identify that hardware failures cause the most damage as the down time is very high. However, the number of failures due to software bugs, misconfigurations, and OS issues together account for around twice the number of hardware failures. The impact of all these software issues is much smaller, however, they can not be ignored. In this study, both hardware and software failures are simulated in DCNs to infer the relationships between the various design parameters like the topology, number of manufacturers, and number of components from the manufacturers. An intuitive result from these works is that a bigger network takes a longer time-to-repair [11].

In [8], the authors show that redundancy is insufficient to tackle massive failures. In [15], a non-parametric Kaplan-Meier survival estimator and a Proportional Hazard model are employed. They are used to find the expected time-to-fail of the components. Then, the important factors that cause frequent failures are assessed. These researches focus on the causes of failures in data centers. They fail to account for the impact of the manufacturers of the components, the arrangement and the interconnection of the components. Massive multiple failure scenarios are not evaluated. In this work, multiple failures due to hardware and software issues are evaluated for the sovereignty analysis.

Our previous work [9] on DCN sovereignty considered HW failures only. Three different approaches- homogeneous, heterogeneous and robustness surfaces were implemented to determine some basic guidelines for the DCN operators. There were four arrangements considered in this work- Random (A_{Ran}), Left-Right (A_{LR}), Left-Right Sequential (A_{LRS}), and Pod-wise (A_{Pod}). In the A_{LRS}, the components in a layer of the DCN are arranged sequentially according to their manufacturers, and this sequence is repeated. Consider Fig. 1, with three manufacturers- blue, purple, and yellow. Each rectangle is a pod. A pod is a small unit made up of a fixed number of Top of the Rack (ToR) and aggregate switches. In the A_{LRS}, the sequence blue-purple-yellow is repeated. From [9], the A_{LRS} performs the best. Therefore, all simulations in this current work use A_{LRS}.

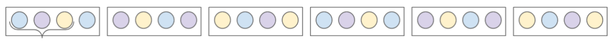


Fig. 1: Left-Right Sequential Arrangement

TABLE I: Infrastructure for all topologies ($32K$ Servers)

Topology	(n_S, n_T, n_A, N_P, N_C)	(L_{ST}, L_{TA}, L_{AC}, L_{CI}, R_A, R_C)
3TLS	(96, 30 8, 11, 6)	(10, 40, 100, 400, -, -)
FT, ABFT	(64, 16, 5, 32, 40)	(10, 40, 40, 40, -, -)
Fb-4P	(44, 48, 4, 15, 4)	(1, 10, 40, 400, 80, 160)
Fb-Fab	(48, 48, 4, 14, 96)	(10, 40, 40, 40, -, -)

III. DCN Modeling

This section discusses the DCN implementation in simulations. The additions to the model in [9] are discussed in detail. Servers, ToR, aggregate, and core switches are arranged in the Top of the Rack architecture throughout this study. The simulation tool has been developed by the authors using Python programming language. The model is split into five parts - input options, macro parameters, micro parameters, software parameters, components of a switch, and output metrics.

A. Input Options

The topologies considered in this work are summarized in [16] and [17]. Leaf and spine switches are present in the lower and upper layers of each pod in the Leaf-Spine topology. Each leaf switch connects to each spine switch in the pod. A 3 Tier Leaf-Spine topology (3TLS) is obtained by adding another upper layer of super spine switches as shown in Fig. 2a. The Fat Tree topology (FT) is a modified Clos-network, shown in Fig. 2b, known for its scalability as all the layers use commodity switches. Every core switch is connected to one aggregate switch in each pod. The AB-Fat Tree topology (AB-FT) is a skewed FT as shown in Fig. 2c. The AB-FT is more robust than the FT, but requires complicated interconnections. There are two types of pods arranged alternately. In type-A pods, each aggregate switch is connected to consecutive core switches. In type-B pods, each aggregate switch is connected to the core layer in steps of fixed length. The Facebook 4-Post topology (Fb-4P) [16] is shown in Fig. 2d. The upper level switches are connected in a ring as seen in Fig. 2d for reliability and efficient load balancing. The Facebook Fabric topology (Fb-Fab) is shown in Fig. 2e. The core layer has 4 planes. Each of the 4 aggregate switches in a pod is connected to all the core switches in a plane. Each core switch is connected to one aggregate switch per pod.

For efficient execution of the simulations, the size of all DCNs in this paper are restricted to $32K$ servers. Depending on the topology, the infrastructure followed is given in Table I. The notation used in this table is explained in Table II. The N_m values range from 2 to 10, for both SW and HW. Throughout this paper, the components are arranged as per the A_{LRS} arrangement as discussed in Section II.

B. Macro parameters

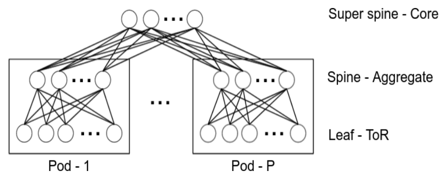
Macro parameters control the major behavior of the DCN. The simulation duration (T_S), number of flows per server per second (N_{SS}), oversubscription ratio (R_{OS}), intra to inter-rack traffic ratio (R_{rack}), and intra to inter-DCN traffic ratio (R_{DCN}) are the macro parameters. T_S is set to $1s$. N_{SS} ranges from 200 to 500 in [18]. [19] is a traffic study in DCNs that reveals up to a $1.6\times$ increase in the N_{SS} between 2015 and 2020. This factor is used to set N_{SS} to 800 for the year 2022. The R_{OS} is the ratio of the bandwidth of all the southbound ports to all the northbound ports [20]. The R_{OS} ranges from $2.5:1$ to $240:1$ [21]. However, the R_{OS} has to be smaller to sustain traffic bursts. In this study, an $R_{OS} = 3:1$ and $R_{OS} = 2:1$ for the ToR and aggregate switches respectively. The Fb-4P is an exception. It follows the architecture described in [16]. Here, $R_{OS} = 10:1$ and R_{OS} of $3:1$ for ToR and aggregate switches respectively. Most of the flows stay within the rack [22]. Therefore, the R_{rack} has been set to $70\%:30\%$. The inter-DCN traffic is considered as a 'special rack' [23] and the R_{DCN} is set to $90\%:10\%$. All these values follow the model in our previous work [9].

TABLE II: Notation used in this work

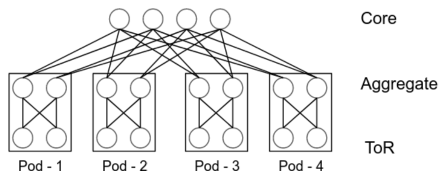
n_S	No.of servers per ToR	n_T	No.of ToRs per pod
n_A	No.of aggregate switches per pod	N_P	No.of pods
N_C	No.of core switches	L_{ST}	Server-ToR link capacity (Gbps)
L_{TA}	ToR-Aggregate link capacity (Gbps)	L_{AC}	Aggregate-Core link capacity (Gbps)
L_{CI}	Core-Inter-DC link capacity (Gbps)	R_A	Aggregate ring capacity (Gbps)
R_C	Core ring capacity (Gbps)	$3TLS$	3 Tier Leaf-Spine Topology
FT	Fat Tree Topology	$AB\text{-}FT$	AB-Fat Tree Topology
$Fb\text{-}4P$	Facebook 4-Post Topology	$FbFab$	Facebook Fabric Topology
T_S	Simulation time	N_{SS}	Flows per server per second
R_{OS}	Oversubscription ratio	R_{rack}	Intra to inter-rack traffic ratio
R_{DCN}	Intra to inter-DCN traffic ratio	TM	Traffic Matrix
L_U	Link utilization	F_s	Types of clusters
DB	Database cluster	FE	Frontend cluster
HW	Hardware	H_m	No.of hardware manufacturers
SW	Software	S_O	Other software developer
S_N	Native software developer	S_{Om}	No.of other software developers
S_{Nm}	No.of native software developers	z_f	Max-flow
Z_C	Connectivity percentage	Z_F	Average max-flow
F_{types}	Types of failures	H_F	Hardware failures
S_{NF}	Native software failures	S_{OF}	Other software failures
$\widetilde{Z_F}$	Lost max-flow	R_{imp}	Ratio of impact of S_{NF} to all SW failures
R_n	Ratio of number of S_N failures to all SW failures	r	Factor by which the impact of S_{OF} is multiplied to get the impact of S_{NF}

C. Micro parameters

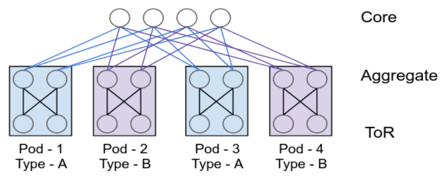
Micro parameters are responsible for the flow characteristics. The flow size, Traffic Matrix (TM), and link utilization



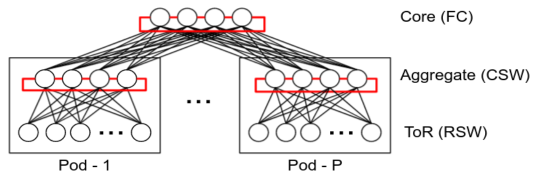
(a) 3 Tier Leaf-Spine (3TLS)



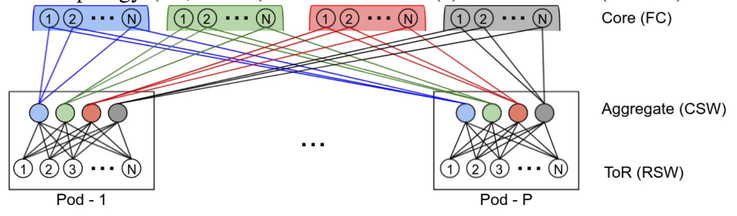
(b) Fat Tree topology (FT, $k = 4$)



(c) AB-Fat Tree (AB-FT)



(d) Facebook 4-Post (Fb-4P)



(e) Facebook Fabric (Fb-Fab)

Fig. 2: Topologies

(L_U) are the micro parameters.

From [18], [24], [25], (i) 80% of all flows are smaller than $10KB$, 95% of all flows are smaller than $1MB$, and 99% of all flows are smaller than $100MB$, (ii) The large flows account for most of the traffic volume. These attributes can be included by a modified Pareto distribution as shown in the Fig. 3a from the simulator. Here, around 45% of all flows are smaller than $1KB$ while around 80% of all flows are smaller than $10KB$.

The volumes of traffic between the source (Y-axis) and destination (X-axis) ToRs are arranged as a matrix called as the TM. The last column is the inter-DC connection. Fig. 3b is a part of the TM from the simulator for a small-sized fat tree topology. The intra-rack traffic is not shown in this figure. The TM is sparse as per [23].

From [18], [22], (i) the L_U increases in the upper layers, (ii) the average ToR-Server L_U is under 1% (iii) 99% of all links are lesser than 10% loaded, (iv) the median ToR-Agg L_U is between $10-20\%$, (v) the busiest 5% of all links are 23–46% loaded. The ECMP algorithm is used for routing the flows in this work.

D. Software Parameters

The DCN has different types of pods or clusters based on the types of servers in the clusters. From [18], the 6 types of clusters (F_s) considered in this paper are Hadoop, Cache, Database (DB), Service, Frontend (FE), and Others. A cluster can be homogeneous, consisting of servers dedicated to a particular type like Hadoop and Cache clusters. They can also be heterogeneous, consisting of different servers. For example, the FE cluster has web servers and Cache servers. Different clusters have different flow characteristics. For example, the Hadoop flows are mostly intra-cluster while the Cache flows are distributed evenly acroos other clusters. The DCN is modeled such that the source-destination locality of the flows match the values present in Table 3 in [18]. Fig. 3c is an example of the Inter-rack TM of an FE cluster, comparable to Fig. 5b in [18].

E. Components of a switch

A switch is made up of HW and SW. The SW for a switch can be from various sources like the switch manufacturer (native software- S_N), DCN operator, and other developers (S_O). The components of the switch as assumed in this study are shown in Fig. 4. Here, H_m, S_{Nm} and S_{Om} denote the number of HW, S_N and S_O developers respectively. Collectively, they are denoted as N_m. The software from the DCN operator is not considered because this is irrelevant to the sovereignty analysis.

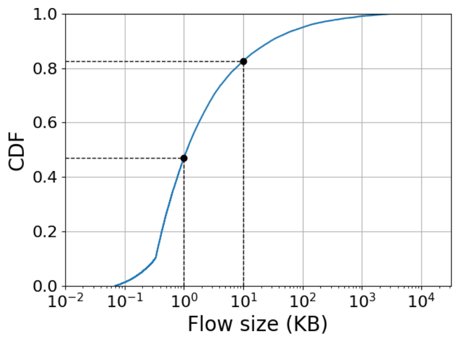
F. Output Metrics

Connectivity percentage (Z_C) and max-flow (z_f) are the primary output metrics considered. Connectivity is the existence of a path between two racks. Note that connectivity does not guarantee successful traffic as there may be other issues like link overload or switch-CPU overload. The Z_C is defined as the ratio of the number of connected communicating pairs of racks to the total number of communicating pairs of racks.

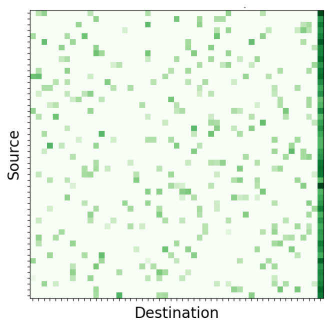
The max-flow min-cut theorem is employed for every source-destination pair of racks to calculate the available z_f. The z_f is highest when there are no failures, decreases as failures are included, and reduces to 0 when the flows can not be routed between the source and destination. When the z_f is small, one or more links are overloaded and blocked. The average max-flow (Z_F) is the average of all the individual z_f. Both the Z_C, and the Z_F are plotted as line graphs for visualization and comparison.

IV. Sovereignty Analysis

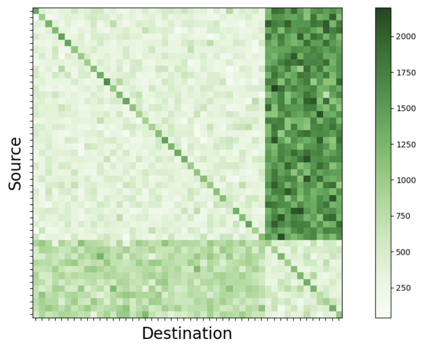
Similar to most existing DCNs, in this analysis, there is no redundancy in the server-ToR links. Therefore, if a ToR fails, all the flows associated with that rack fail. Hence, the failures of ToRs is not of interest in this work. In this section, the two methods for analyzing the sovereignty of a DCN are detailed. The homogeneous analysis discussed in [9] is not possible here because the number of combination for failure



(a) Flow Size-CDF



(b) ToR-TM for flows



(c) Inter-rack TM for FE cluster

Fig. 3: Images from the simulator

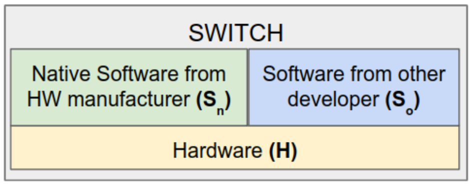

SWITCH
Native Software from HW manufacturer (S_n)
Software from other developer (S_o)
Hardware (H)

Fig. 4: Components of a switch

scenarios is very large considering both the hardware and software manufacturers.

A. Failures

There are 3 types of failures (F_{types})- hardware failures (H_F), other software failures (S_{OF}), and native software failures (S_{NF}). For H_F, it is considered as a complete switch failure. For S_{OF} and S_{NF}, some flows are affected. The flows that are affected can be based on any criterion. In this paper, we assume the criterion to be the source of the flows. We assume that the S_{OF} and S_{NF} can affect the flows from one (or more) of the 6 different types of clusters. This assumption is made because there are no previous studies that explain the flows that are affected due to a software failure in a switch. The simulator can also accommodate S_{OF} and S_{NF} impacts based on other criteria like destination and flow size.

There is a chance that the software failures may not cause the flows to be abruptly dropped. They may cause additional latency and degrade the performance. However, in our work, we consider the flows that can be routed (even with a latency) as successful flows. We only focus on the flows that are completely blocked.

B. Heterogeneous analysis

This analysis determines the maximum number of components that can be used from a single manufacturer, if the DCN is planned to tolerate the complete failure of one HW, S_O, and S_N manufacturer simultaneously. For each topology, one Z_F line graph is generated for HW, S_O, and S_N failures each.

For S_{OF}, flows from one of the 6 different types of clusters are affected. It can be argued that the failure can affect flows

Algorithm 1: Evaluation of failures

```
1  Macro and micro parameters are loaded.
2  foreach topology do
3      Generate topology as a graph.
4      Generate flows with source, destination and size.
5      Generate the TM.
6      if HW failure then
7          Failing switches are removed from the
             topology Perform ECMP routing. Save
             $Z_C[HW]$ and $Z_F[HW]$.
8          $\widetilde{Z_F}[HW]$
9      end
10     else if $S_O$ failure then
11         foreach $f_s \in F_s$ do
12             Assume flows with source $f_s$ can not be
                 routed via switches with SW failure.
13             Perform ECMP routing.
14             Save $Z_C[S_O][f_s]$ and $Z_F[S_O][f_s]$,
                 $\widetilde{Z_F}[S_O][f_s]$.
15         end
16         $Z_F[S_O](ave) = \frac{1}{|F_s|}\sum_{f_s} Z_F[S_O][f_s]$ .
17         $Z_F[S_O](max) = max(Z_F[S_O][f_s])$ .
18     end
19     else
20         $Z_{F-all} = \{Z_F[S_O](ave), Z_F[S_O](max)\}$
21         foreach $z \in \{Z_C, Z_F\}$ do
22             foreach $t \in \{ave, max\}$ do
23                 foreach $r \in \{0.5, 0.75, .., 3\}$ do
24                     // Sensitivity analysis.
25                     $z[S_N][t] = r \times z[S_O][t]$ .
26                 end
27             end
28         end
29     end
30 end
```

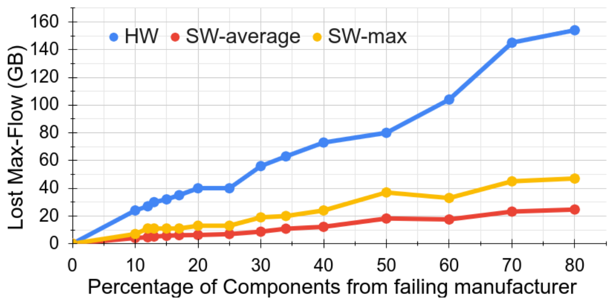


Fig. 5: $\widetilde{Z_F}[HW]$ for Fb-Fab

from more clusters. However, as a case study, only one source cluster is considered. There are no previous studies that explain the relative impact of S_{NF} with respect to that of S_{OF}. Depending on the severity of the S_N bug, the impact will vary.

Here, the impact of S_{NF} is taken as a multiple (r) of the impact of the S_{OF}. That is, Z_F for S_{OF} is calculated first. S_{NF} is also a software failure. Hence, S_{NF} will also have a similar pattern of impact on the network. However, the severity of the impact will vary. This severity is modeled using the variable r. Therefore, the Z_F for S_{NF} is calculated as a multiple of the Z_F for S_{OF}. This is shown in line 25 of Algorithm 1. Since there is no previous data on the variable r, a sensitivity analysis ($r \in \{0.5, 0.75, .., 3\}$) is necessary.

The failures in F_{types} are evaluated one by one as per the Algorithm 1. This algorithm is run for different percentages of components from the failing manufacturer ($N_C = \{0, 10, 12, 13, 15, 17, 20, 25, 30, 34, 40, 50, 60, 70, 80\}$). These numbers are chosen so that it is easier to choose S_{Nm} and S_{Om}. For further calculations, max-flow is used because they give more meaning. All the Z_{F-all} values denote the 'available max-flow' in a particular case. Subtracting these values from the available max-flow in the no-failure scenario, gives the 'Lost max-flow' ($\widetilde{Z_F}$). To find the required N_m to run the DCN, it is easier to make calculations based on how much max-flow the DCN loses due to the failures.

Consider the example of hardware failures in an Fb-Fab shown in Fig. 5. The X-axis is the percentage of components from the failing manufacturer. As N_C increases, the $\widetilde{Z_F}[HW]$ increases. Note that the impact of the HW failure is far greater than the SW failures. This is because the HW failure renders the switch useless, whereas the SW failures only affect some flows in the switch. As expected, the worst-case scenario of failures represented in the $SW(max)$, causes more loss than the average case in $SW(ave)$.

Section IV-4 in [9] explains the current traffic demands and expected traffic demands in 5 years as a case study. The same is applied here. For the current traffic demands, 66% of the best case Z_F is required for the effective working of the DCN. This can be restated as the DCN can tolerate upto 34% of $\widetilde{Z_F}$. Therefore, the threshold, $\widetilde{Z_{Ft}}$ is 34%. Similarly, in the future, $\widetilde{Z_{Ft}}$ is 24%.

After executing algorithm 1 for different cases, it is essential to find N_m needed, such that the DCN can tolerate the complete failure of one HW, S_O, and S_N manufacturer simultaneously. For this, an optimization problem is formulated. Let b_{ij} be the binary decision variable where,

$$\forall i \in F_{types}, j \in N_C, b_{ij} = \begin{cases} 1, & \text{if j\% is alowed} \\ 0, & \text{otherwise} \end{cases}$$

Let $\widetilde{Z_{Fij}}$ be the $\widetilde{Z_F}$ for failure type i and percentage of failing components j. There are 3 constraints. Constraint 1 is for choosing only one N_C value. Constraint 2 ensures equal percentages for HW and N_S manufacturers because HW and N_S can not be from different manufacturers on the same switch. They have to be equal. Constraint 3 is the important limitation. This constraint limits the losses due to the different F_{types} to a value lower than the threshold $\widetilde{Z_{Ft}}$, that can be tolerated by the DCN.

$$\sum_{j \in N_C} b_{ij} = 1, \forall i \in F_{types}, \tag{1}$$

$$b_{HW,j} = b_{S_N,j}, \forall j \in N_C, \tag{2}$$

$$\sum_{\substack{j \in N_C \\ i \in F_{types}}} b_{ij} \times \widetilde{Z_{Fij}} \leq \widetilde{Z_{Ft}}. \tag{3}$$

The primary objective function is to minimize H_m. This is because, SW sources are abundant in today's market. However, the supply of HW is limited. Therefore, a DCN operator would want to have the least H_m in the DCN for easier procurement and management of components. To minimize H_m, the upper limit on the number of components that can be purchased from a single manufacturer must be maximized. For example, if the percentage of components that can be purchased from a single manufacturer is 25%, then the operator needs 4 manufacturers. But, if the percentage of components that can be purchased from a single manufacturer raises to 34%, then the operator needs only 3 manufacturers to build the DCN. Therefore, we attain the objective of minimizing H_m, by maximizing the number of components that can be purchased from one manufacturer, as seen in Eq. O1.

Another objective function (O2), which minimizes the total number of manufacturers, has also been tested. In this objective function, the operator tries to minimize the sum of all the number of manufacturers instead of minimizing only the number of hardware manufacturers. S_{Nm} is not in the objective function because it is already equal to H_m as per 3. Both these objective functions were tested out separately.

$$max(\sum_{j \in N_C} j \times b_{HW,j}) \tag{O1}$$

$$max(\sum_{j \in N_C} (j \times b_{HW,j} + j \times b_{S_O,j})) \tag{O2}$$

The results of this optimization problem is used to find the required number of manufacturers as discussed in Sec. V-A.

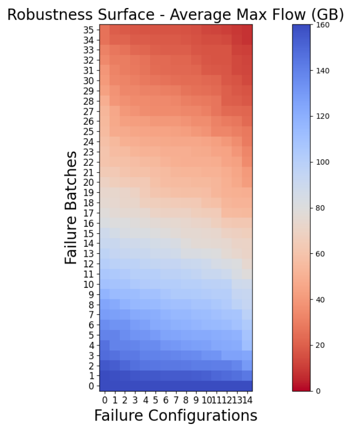


Fig. 6: Z_F - FbFab, 5 HW and S_O manufacturers, $R_n = 0.8, R_{imp} = 3$

C. Robustness Surfaces

The Robustness Surface (RS) [26] is a tool used to compare the performance of different networks under failures. The Fb-Fab with $5HW$ and S_O manufacturers is shown in Fig. 6 as an example. The Failure Configurations (FC) are along the X-axis. An FC is the realization of failures in a specific order. Each FC is a different order. The failure batch count of is on the Y-axis. Here, the components are purchased homogeneously with different failure rates.

Let the m^{th} manufacturer of failure type i from F_{type} be denoted by N_{im}. Here, N_{i1} has $2\times$ the failure rate of N_{i0} and so on. As an example, Fig. 6 shows the Z_F robustness surface for the Fb-Fab topology with 5 HW manufacturers and S_O manufacturers. In the 1^{st} row for Batch-0, there are no failures. From the 2^{nd} row, for every row, there are 5% of components experiencing failures incrementally. Out of these 5%, one third are HW failures, and two-thirds are SW failures. This ratio of $1:2$ between HW and SW failures is taken from [8], [11], [13]. In the SW failures, R_n is the ratio of number of S_N failures to all SW failures ($\frac{S_N}{S_N+S_O}$). Similarly, R_{imp} is the ratio of impact of S_N failures to all SW failures. R_n and R_{imp} are not discussed in any previous works. Therefore, a sensitivity analysis is done considering $R_n \in \{0.2, 0.4, 0.6, 0.8\}$ and $R_{imp} \in \{1, 2, 3\}$. Note that the R_{imp} in this analysis is similar to the r in Section IV-B, but not the same. Here, R_{imp} gives the number of switches that have the same flows being affected. The Z_C and Z_F are noted for each simulation and it is repeated for every column. Finally, the rows are arranged in descending order. From Fig. 6, when more failures are present, the Z_F decreases.

V. Results

There are 2 sections- Heterogeneous and RS results. The results for for the Fb-Fab topology are shown as examples in this section because the Fb-Fab topology is the most recent topology that is used widely in data center networks. The simulation on Python took approximately 60 seconds to generate the output for one topology, one value of N_C, and one value of r. The time taken varies slightly for different topologies and percentage of failing components due to the change in the size of the DCN. The solution for the optimization problem was generated in approximately 0.4 seconds. Note that the time taken for simulation is not the objective of this analysis and it does not impact the results obtained.

A. Heterogeneous Analysis

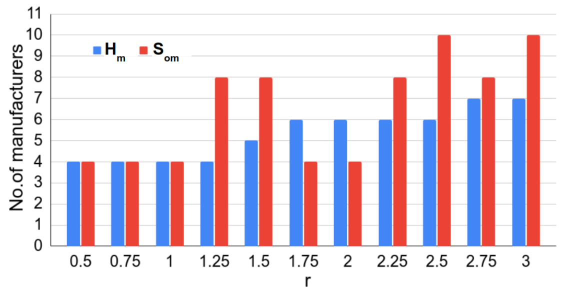
Solving the optimization problem gives the required N_m. Repeating it for different topologies, r, and, traffic demands, the heterogeneous analysis produces the following results as seen in Fig. 7.

In Fig. 7a, N_m are compared for different values of r. r determines the impact of S_N failures. As r increases, the impact increases. Hence, N_m increases. The Z_F saved by increasing 1 H_m is much higher than 1 S_O, because, the impact of HW failures are much higher. Between $r = 1$ and $r = 1.25$, the difference in $\widetilde{Z_F}$ is small. Hence, it can be managed by increasing S_{Om}. Between $r = 1.5$ and $r = 1.75$, the difference in $\widetilde{Z_F}$ is large. Thus, H_m has to be increased. The $\widetilde{Z_F}$ saved by increasing H_m allows more flexibility for S_{Om} to reduce. Therefore, there is an inverse relationship between H_m and S_{Om}. Increasing one of them can compromise the shortage in another.

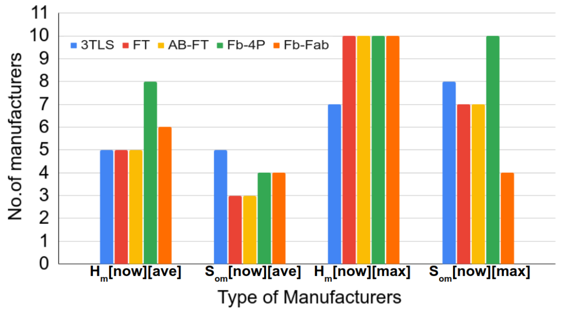
In Fig. 7b, the different topologies are compared as an example for $r = 2$. Along the X-axis, the different manufacturers are arranged with the first two for the average case and the next two for the worst case as per Algorithm 1. The N_m required for the worst case (max) is more than that for the average case, because, when constraints are strict, more N_m is required. Most topologies have reached $H_m = 10$ for the max case. This shows that under severe constraints, many configurations will require H_m more than 10, which is infeasible in this analysis because the upper limit for H_m has been set to 10. In the worst-case, 3TLS performs best followed by Fb-Fab, owing to their robustness and redundancy. Fb-4P performs the worst.

In Fig. 7c, N_m for different traffic demands is calculated. In this example for an Fb-Fab, $r = 1$, more N_m is required to cope with the higher traffic demands in the future. Table III summarizes the findings of this analysis. The above results are for the objective O1. Evaluating the results for objective O2, similar trends were observed; the only difference being H_m and S_{Om} were being minimized together. So some results had a higher required H_m, but had the least $H_m + S_{Om}$ requirement.

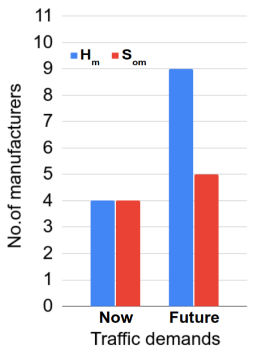
For the sake of completion and checking correctness, one more objective function to minimize the number of software manufacturers was also tested. In all the above optimizations, the required H_m, S_{Om}, and S_{Nm} varied, but feasibility did not vary. That is, if an optimization problem for a particular topology was infeasible in the first objective function O1, it was also infeasible in the other objective functions.



(a) Fb-Fab, r comparison for the average case.

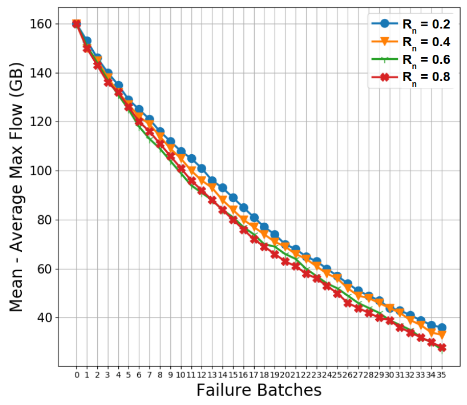


(b) $r = 2$, Topology and Average-Max comparison

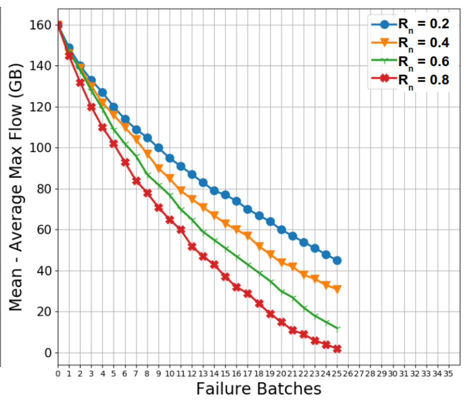


(c) Fb-Fab, $r = 1$, Now-Future comparison

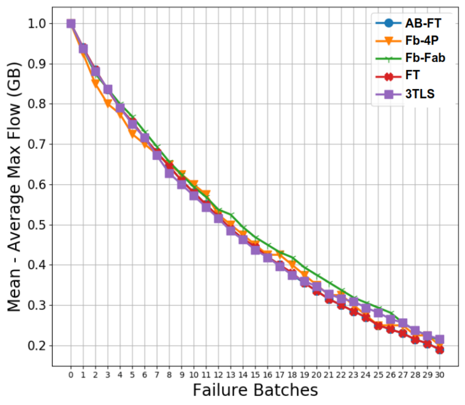
Fig. 7: Results from the Heterogeneous analysis, $H_m = 5, S_{Om} = 5$



(a) Fb-Fab, $R_{imp} = 2$, R_n comparison



(b) $r = 2$, Topology, Average-Max comparison



(c) Fb-Fab, $r = 1$, Now-Future comparison

Fig. 8: Results from the Robustness Surfaces analysis, $H_m = 5, S_{Om} = 5$

TABLE III: Heterogeneous Analysis - Summary

Scenario	3TLS	FT	AB-FT	Fb-4P	Fb-Fab
Z_C	+++	++	++	-	++
Z_F	No big difference				
H_m, S_{Nm}, and S_{Om} required under strict traffic constraints	+++	+	+	-	++
Cost	-	+	+	++	+++
Ease of Implementation	+++	++	-	++	++

B. Robustness Surfaces

Since the example in Fig. 6 is difficult to compare with other RSs, the values along each row are averaged to get a new graph as in Fig. 8. The Fig. 8a is an example of the Fb-Fab with $R_{imp} = 2$. As R_n increases, the number of S_N failures increases. Since the impact of S_N is $2\times$ that of S_O failures, the Z_F decreases more. Therefore, as R_n increases, Z_F decreases. In Fig. 8b, the same graph is plotted for $R_{imp} = 10$. Hence, the impact of S_N is $10\times$ that of S_O failure. Therefore, Z_F decreases even more than in Fig. 8a. The number of failure batches is reduced in the latter because, the DCN can not function with more batches when R_{imp} is so high. In Fig. 8c, $R_n = 0.4$ and $R_{imp} = 5$. This example is to compare topologies. There is no clear winner in the topologies from the RS analysis.

From this analysis, when R_n increases, or R_{imp} increases, or any other constraint is introduced, the Z_F decreases. To handle this decrease, more N_m will be needed. This agrees with the previous findings in Heterogeneous analysis in Sec. V-A.

VI. Conclusion

From the above results, the following guidelines can be inferred for the DCN operators.

(G1) In small DCNs (less than 5000 servers), Leaf-Spine can be used. In larger DCNs, a Clos-network-based topology is needed. The increase in number of cables, increase in the capacity of switches at higher layers, and complexity of physical connections make the Leaf-Spine based topologies expensive and unsuitable for large DCNs.

(G2) For strict constraints (like high R_n, high R_{imp}, *max* case, future demands) more manufacturers are required to combat massive failures. For very high constraints, the topologies can become infeasible.

(G3) The severity of SW bugs plays an important role in determining the number of required manufacturers.

(G4) There exists an inverse relationship between H_m and S_{Om}. If H_m increases, there is space for S_{Om} to decrease and vice versa.

(G5) Traffic demands in the future are more. Therefore, more H_m and S_{Om} are needed.

(G6) Irrespective of the objective functions for optimization, the feasibility of the DCN remains the same.

Data centers were taken as an appropriate first step to study sovereignty in networks. Two different approaches were implemented to give an idea of how a sovereignty analysis can be conducted in a network. Different topologies were modeled analyzed for their robustness under massive failure scenarios based on the number of HW and SW manufacturers used to build the DCN. The results from these analyses give general guidelines to DCN operators to build a sovereign and reliable DCN. This analysis can bee extended in the future to other wired networks like optical fiber networks, and core networks, and later to wireless networks.

References

[1] "*US* telcos ordered to 'rip and replace' huawei components," *BBC News*, December 2020. [Online]. Available: https://www.bbc.com/news/business-55269879

[2] L. Kelion, "Huawei *5G* kit must be removed from *UK* by 2027," *BBC News*, July 2020. [Online]. Available: https://www.bbc.com/news/technology-53403793

[3] "*EU* sanctions against *Russia* explained," *Council of the EU and the European Council*, July 2022. [Online]. Available: https://www.consilium.europa.eu/en/policies/sanctions/restrictive-measures-against-russia-over-ukraine/sanctions-against-russia-explained/

[4] M. J. Loveridge, G. Remy, N. Kourra, R. Genieser, A. Barai, M. J. Lain, Y. Guo, M. Amor-Segan, M. A. Williams, T. Amietszajew *et al.*, "Looking deeper into the galaxy (note 7)," *Batteries*, vol. 4, no. 1, p. 3, 2018.

[5] A. Rudolph, "What is log4j and why did the government of canada turn everything off?" 2022.

[6] J. Luszcz, "Apache struts 2: how technical and development gaps caused the equifax breach," *Network Security*, vol. 2018, no. 1, pp. 5–8, 2018.

[7] V. Singh and P. Chaudhary, "Y2k38: The bug," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 2, 2012.

[8] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: measurement, analysis, and implications," in *Proceedings of the ACM SIGCOMM 2011 Conference*, 2011, pp. 350–361.

[9] S. Janardhanan and C. Mas Machuca, "Modeling and evaluation of a data center sovereignty," in *International Conference on the Design of Reliable Communication Networks (DRCN)*, 2022.

[10] J. Edler, K. Blind, R. Frietsch, S. Kimpeler, H. Kroll, C. Lerch, T. Reiss, F. Roth, T. Schubert, J. Schuler *et al.*, "Technology sovereignty: from demand to concept," Perspectives-Policy Brief, Tech. Rep., 2020.

[11] J. Meza, T. Xu, K. Veeraraghavan, and O. Mutlu, "A large scale study of data center network reliability," in *Proceedings of the Internet Measurement Conference 2018*, 2018, pp. 393–407.

[12] R. Potharaju and N. Jain, "When the network crumbles: An empirical study of cloud network failures and their impact on services," in *Proceedings of the 4th annual Symposium on Cloud Computing*, 2013, pp. 1–17.

[13] X. Wu, D. Turner, C.-C. Chen, D. A. Maltz, X. Yang, L. Yuan, and M. Zhang, "Netpilot: Automating datacenter network failure mitigation," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, 2012, pp. 419–430.

[14] S. Choi, B. Burkov, A. Eckert, T. Fang, S. Kazemkhani, R. Sherwood, Y. Zhang, and H. Zeng, "Fboss: building switch software at scale," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 342–356.

[15] R. Singh, M. Mukhtar, A. Krishna, A. Parkhi, J. Padhye, and D. Maltz, "Surviving switch failures in cloud datacenters," *ACM SIGCOMM Computer Communication Review*, vol. 51, no. 2, pp. 2–9, 2021.

[16] N. Farrington and A. Andreyev, "Facebook's data center network architecture," in *2013 Optical Interconnects Conference*. Citeseer, 2013, pp. 49–50.

[17] B. Lebiednik, A. Mangal, and N. Tiwari, "A survey and evaluation of data center network topologies," *arXiv preprint arXiv:1605.01701*, 2016.

[18] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 123–137.

[19] C. V. N. Index, "Forecast and methodology, 2015–2020," *White paper*, pp. 1–41, 2016.

[20] R. R. Reyes and T. Bauschert, "Infrastructure cost comparison of intra-data centre network architectures," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2018, pp. 1–7.

[21] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: A scalable and flexible data center network," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, 2009, pp. 51–62.

[22] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 267–280.

[23] Z. Hu, Y. Qiao, and J. Luo, "Atme: Accurate traffic matrix estimation in both public and private datacenter networks," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 60–73, 2015.

[24] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 92–99, 2010.

[25] ——, "Microte: Fine grained traffic engineering for data centers," in *Proceedings of the seventh conference on emerging networking experiments and technologies*, 2011, pp. 1–12.

[26] M. Manzano, F. Sahneh, C. Scoglio, E. Calle, and J. L. Marzo, "Robustness surfaces of complex networks," *Scientific reports*, vol. 4, no. 1, pp. 1–6, 2014.